PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

A1

(51) International Patent Classification 6 : H04N 7/14	A1	(11) International Publication Number: WO 98/11724
		(43) International Publication Date: 19 March 1998 (19.03.98)

(21) International Application Number: PCT/US97/17111

(22) International Filing Date: 11 September 1997 (11.09.97)

(30) Priority Data:

60/026,056	13 September 1996 (13.09.96)	US
08/790,132	29 January 1997 (29.01.97)	US

(71) Applicant: SARNOFF CORPORATION [US/US]; CN 5300, 201 Washington Road, Princeton, NJ 08540 (US).

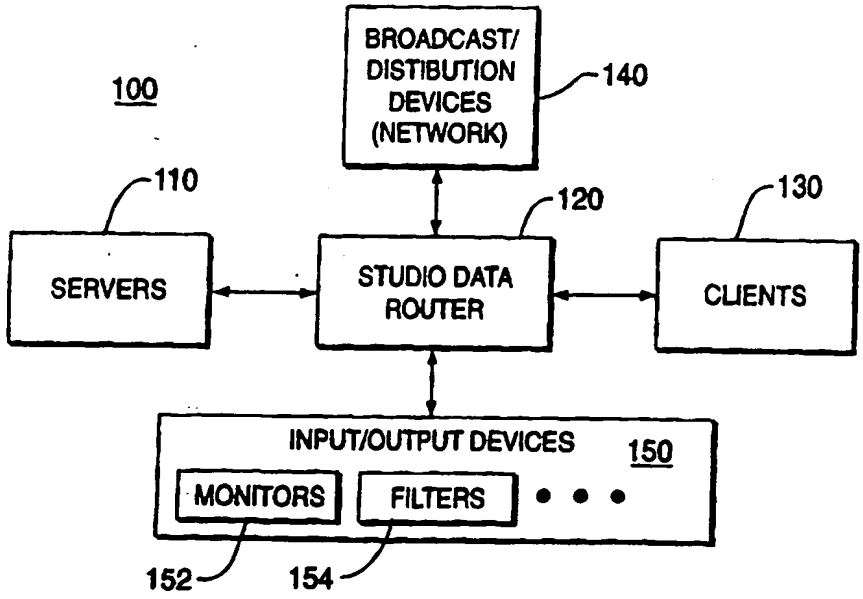
(72) Inventors: WARD, Christopher; 144 Essex Avenue, Glen Ridge, NJ 07028 (US). WINE, Charles, M.; 144 Hamilton Avenue, Princeton, NJ 08540 (US). HOMAN, David, B.; 21 Columbia Avenue, Hopewell, NJ 08525 (US).

(74) Agents: SILVERIO, John, V. et al.; Sarnoff Corporation, CN 5300, 201 Washington Road, Princeton, NJ 08540 (US).

(81) Designated States: BR, CA, CN, JP, KR, MX, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: COMMAND AND CONTROL ARCHITECTURE FOR A DIGITAL STUDIO



(57) Abstract

A command and control architecture for a compressed digital television broadcast studio (100). The architecture includes provisions for session management, real-time control of studio activities, the use of proxy objects to establish communications with objects not providing a high level of control, control of studio filter devices (154) and stream management. The invention includes an object architecture particularly suited to the digital studio.

This Page Blank (uspto)

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece			TR	Turkey
BG	Bulgaria	HU	Hungary	ML	Mali	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MN	Mongolia	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon			PL	Poland		
CN	China	KR	Republic of Korea	PT	Portugal		
CU	Cuba	KZ	Kazakstan	RO	Romania		
CZ	Czech Republic	LC	Saint Lucia	RU	Russian Federation		
DE	Germany	LI	Liechtenstein	SD	Sudan		
DK	Denmark	LK	Sri Lanka	SE	Sweden		
EE	Estonia	LR	Liberia	SG	Singapore		

3NSDOCID: <WO___9811724A1_I_>

This Page Blank (uspto)

COMMAND AND CONTROL ARCHITECTURE FOR A DIGITAL STUDIO

The invention was made with Government support under Contract No. 70NANB5H1174. The Government has certain rights in the invention.

CROSS-REFERENCE TO RELATED APPLICATIONS

This non-provisional U.S. national application, filed under 35 U.S.C. 111(a) claims, under 35 U.S.C. 119(e), the benefit of the filing date of provisional U.S. application no. 60/026,056 filed under 35 U.S.C. 111(b) on September 13, 1996.

The present invention relates to a command and control architecture and, more particularly, to a command and control architecture and concomitant method for management of digital bitstreams in a digital television broadcast studio.

BACKGROUND OF THE INVENTION

The proliferation of digital information has created a new HDTV (High Definition Television) industry employing the concept of a "digital studio". In brief, a digital studio is an environment or system having numerous components where various sources of digital information can be selectively accessed and manipulated (in real time or in delay mode) by multiple clients. For example, a customer can schedule a movie around a "live" broadcast, such that the movie is forwarded to the customer's television by a provider at a particular time, where the movie is paused to forward the "live" broadcast and then resume from the point where the movie was interrupted. Furthermore, digital information is often converted into compressed bitstreams, creating issues such as latency and data rate requirements. Thus, the command and control architecture necessary to integrate the various digital studio components presents a unique challenge.

Currently, there is a Digital Storage Media Command and Control (DSM-CC) architecture (ISO/IEC 13818-6, hereby incorporated by reference), which is a specification of a set of protocols for managing the functions and operations of ISO/IEC 11172 Motion Pictures experts Group (MPEG-1) and ISO/IEC 13818 (MPEG-2) bitstreams. More specifically, the DSM-CC protocol set is particularly

BNSDOCID: <WO__9811724A1_I_>

designed to provide command and control to a system architecture for the delivery of MPEG video streams from a server to a client, e.g., a set-top box. However, since it is anticipated that the numerous components of the digital television broadcast studio will be under distributed control, the current DSM-CC protocol set lacks the flexibility and features that are contemplated for a digital studio.

SUMMARY OF THE INVENTION

A command and control architecture for a compressed digital television broadcast studio. The architecture includes provisions for session management, real-time control of studio activities, the use of proxy objects to establish communications with objects not providing a high level of control, control of studio filter devices and stream management. The invention includes an object architecture particularly suited to a digital studio.

BRIEF DESCRIPTION OF THE DRAWING

The teachings of the present invention can be readily understood by considering the following detailed description in conjunction with the accompanying drawings, in which:

Fig. 1 illustrates a block diagram of a digital studio;

Fig. 2 illustrates a detailed block diagram of one configuration of a digital studio;

Fig. 3 is a graphical illustration of an embodiment of a studio class hierarchy (studio object world) for the devices or components within a digital studio;

Fig. 4 illustrates a studio object directory structure;

Fig. 5 is a graphical illustration of DS-CC dialogue for a studio including a proxy - controlled device;

Fig. 6 is a timing diagram for the DS-CC dialogue shown in Fig. 5;

Fig. 7 is a graphical illustration of a DS-CC dialogue for a studio including two proxy - controlled devices;

Fig. 8 is a timing diagram for the DS-CC dialogue shown in Fig. 7;

Fig. 9 is a graphical illustration of a DS-CC dialogue where the video

NSDOCID: <WO__9811724A1_I_>

stream passes through a transcoder filter;

Fig. 10 is a timing diagram for the DS-CC dialogue shown in Fig. 9;

Fig. 11 is a graphical illustration of a DS-CC dialogue for a studio including two servers connected to a switcher;

Fig. 12 is a timing diagram for the DS-CC dialogue shown in Fig. 11;

Fig. 13 is a graphical illustration of a DS-CC dialogue for storing video from a server to a client;

Fig: 14 is a timing diagram for the DS-CC dialogue shown in Fig. 13;

Fig. 15 is a graphical illustration of a DS-CC dialogue showing the distribution of video streams to multiple recipients and shared clients;

Fig. 16 is a timing diagram for the DS-CC dialogue shown in Fig. 15; and

Fig. 17 is another embodiment of a timing diagram for the DS-CC dialogue shown in Fig. 15

DETAILED DESCRIPTION

The present invention provides a command and control architecture to address issues including, but not limited to, session management, real-time control, proxy device control and stream management for a digital studio, such as the one shown in FIG. 1.

FIG. 1 illustrates a block diagram of a digital studio 100 comprising a studio data router 120, a plurality of servers 110, a plurality of broadcast/distribution devices or networks 140, a plurality of input/output devices and a plurality of clients 130. The digital studio 100 permits the clients 130 to selectively access and/or control the various studio resources, where the clients may include, but is not limited to, a set-top box, a computer or a storage device. In fact, since it is contemplated that the digital studio will be under distributed control, other studio components may serve as clients as well, e.g., the various servers in the digital studio.

The digital studio system components are physically connected to one another through a central switch called the studio data router 120. The studio data router 120 processes the video streams which travel through the digital studio, and also provides services, such as command and control services.

BNSDOCID: <WO__9811724A1_I_>

The studio components may include broadcast/distribution devices or networks 140 which may include, but is not limited to, a satellite distribution network, a broadcast network or a local "live-feed" network and various servers 110 holding a large quantity of stored digital information. Furthermore, the studio components may include input/output devices such as monitors 152 and various filters 154, e.g., transcoders, converters, codecs, interface devices and switchers. Each of the various studio components may incorporate the necessary hardware (e.g., one or more processors, computers or workstation) to store or implement software routines or objects. In fact, the various studio components may incorporate other direct signal connections (not shown) among themselves to effect direct communication with each other.

One low level transport protocol which can be used within the digital studio and the present command and control architecture is the Asynchronous Transfer Mode (ATM), which is well suited to provide multiple data streams to a single point. However, those skilled in the art will realize that other suitable low level transport protocols can be used. Thus, disclosed herein is a command and control architecture for the environment of FIG. 1 or other equivalent digital studio configurations, e.g., as illustrated in FIG. 2.

The command and control architecture for the digital studio is based on the recognition that broadcast studios using HDTV compressed bitstream technology will include intelligent cooperating system resources/components under distributed control. As such, the studio environment of today, which is modeled on a hierarchy of control emanating from "play-list managers", where data is move between studio components in frame synchronized manner, is replaced by the present architecture in which studio video servers, live feeds, monitors, transcoders and switchers are under the distributed control of software objects accessed through workstation user interfaces. Included in this object world will be a "PlayList" class which provides functions consistent with current play-list managers. A programming paradigm well suited to this environment is the concept of distributed object oriented computing, such as the Common Object Request Broker Architecture (CORBA) model. For a detailed discussion on CORBA, see Understanding CORBA, by Otte et al., Prentice Hall PTR, 1996.

INSDOCID: <WO__9811724A1_I_>

The command and control architecture, called "Digital Studio Command and Control" (hereinafter DS-CC), is an improvement over DSM-CC, in that DS-CC includes provisions for the studio environment. More specifically, data streams are extended to make provision for time control, recording and other studio related considerations and functions. Second, the concept of "proxies" is disclosed to permit all studio components to participate in the studio command and control process. Finally, the management of resources is extended to include components that are used to manage streams within the studio context (e.g., transcoders and switchers). As such, although the command and control architecture is described in conjunction with the DSM-CC protocol, those skilled in the art will realize that it is not so limited.

FIG. 3 is a graphical illustration of an embodiment of a studio class hierarchy 500 (studio object world) for the devices or components within a digital studio, e.g., as illustrated in FIG. 2. A class is a template for defining the methods and variables for a particular type of object. Those skilled in the art will realize that the terms "object" and "class" are used in the context of object-oriented computing.

The root of the object class hierarchy or "Naming Context", (conforming with DSM-CC terminology), is a base class 505. Within the base class are the object classes "Devices" 510, "Servers" 520, "Studios" 530 and a variety of software resources 540, e.g., PlayListManagers ("PlayListMgrs"). The Devices class contain three broad categories of components: sources, filters and sinks. More specifically, Devices class 610 contain classes "LiveFeeds" 512 (sources), "Filters" 514 and "Transmitters" (sinks) 516, where the Filter class 514 contains classes "Switchers" 518 and "Transcoders" 519. Similarly, Servers class 520 contains classes "PTAServers" (Play-to-Air) 522, "Network Server" and "ProdProgServer" (Production Program Server) 524. Finally, each class may incorporate numerical designations to specify individual instances of particular studio devices, e.g., "liveFeeds-1", "Switchers-1", "Transcoder-1", and etc.

Furthermore, in each class, a "factory" (also known as "object constructor" for creating new objects) is provided which is used to construct and manage the class in question. To illustrate, for the Device class 510, it is assumed that a

BNSDOCID: <WO___9811724A1_I_>

"DeviceFactory" exists and is designed to create Devices objects. The location of the objects derived from these classes is accessible through a studio wide directory structure 600 as illustrated in FIG. 4.

FIG. 4 illustrates a studio object directory structure 600 with a root directory which is a gateway service interface 605. The gateway service interface provides a directory of services and enables a Client to attach to a service domain. In one embodiment, the gateway service interface is the DSM-CC Service Gateway.

Within the gateway service interface 605 are the class directories "Devices" 610, "Servers" 620, "Studios" 630 and "Software Resources" 640 (SWResources). Security within this object hierarchy is provided using the security provisions within the DSM-CC. The general characteristics of the classes are described below.

The "Devices" directory 610 corresponds to all components within the studio that may be manipulated by other objects (i.e., are addressable). Some examples of devices are the general classes "LiveFeeds" 612, "Filters" 614, "Transmitters" 616 and "Monitors" (not shown). In turn, the "Live Feeds" directory 612 contains classes of objects that correspond to video sources that are being delivered from a location which does not provide the flexibility of a server. Examples of live feeds are cameras, and satellite feeds from other locations. In general, this class of object will have only limited control functionality.

The Filters directory 614 contains classes of objects through which video streams pass but are not considered end points. Examples of these filters include transcoders and switchers where their objects are contained respectively in "Switchers" directory 618 and "Transcoders" directory 619.

The "Transcoders" directory 619 contains classes of objects that manipulate the characteristics of video streams, e.g., the picture frame rate or the picture aspect ratio. The "Switchers" directory 618 contains classes of objects that take several video streams as sources and permit the switching (or splicing) between streams to generate an output stream different from any one of the inputs.

The "Transmitters" directory 616 contains classes of objects that correspond to video sinks. Example transmitters include, but is not limited to, the broadcast

INSDOCID: <WO___9811724A1_I_>

transmitter and satellite transmitter.

The "Monitors" directory (not shown) contains classes of objects that correspond to studio monitors. These are considered a very general resource and will be used by many of the software resources.

The "Servers" directory 620 contains the names of studio servers within the studio. In one example, the Servers directory 620 contains three servers, a Play to Air Server ("PTAServer") 622, a Network Program Server ("NPServer") and a Production Program Server ("PPServer") 624. The servers store video objects and respond to requests from other objects to deliver and/or store video streams to the requested locations.

The "Studios" directory 630 contains the names of studio objects. Each Studios directory contains the name "Local" which corresponds to the local studio itself.

The "SWResources" directory 640 contains the names of object classes that are used within the studio environment for the purposes of studio management. One such type of class is the "PlayListMgrs" class. The "Play List Managers" directory 642 provides objects corresponding to particular types of play list, e.g., "PlayListMgr-1". A PlayListMgrs object is an object that manages a play list by initiating and switching between video streams to provide a single output stream.

Namely, FIG. 4 can be perceived as a naming space for all the different digital studio components, where the functionality of the different components of the digital studio are distributed. As such, traversing the naming space is equivalent to moving physically from one component to another component.

The described command and control architecture provides for real-time stream control. The DS-CC stream commands specify an absolute time at which the command is to be initiated by the object receiving the request. For example the DS-CC "Play" command is "PlayAt(time)", where time is based on some studio accurate notion of time. Other DS-CC stream commands (e.g., Stop, Pause, etc.) can also be extended to specify an absolute time of occurrence in a similar manner.

A detailed discussion of the present DS-CC architecture is now presented below with reference to FIGs. 5-17. These figures illustrate the application of the

BNSDOCID: <WO__9811724A1_I_>

DS-CC architecture to different operating scenarios within a digital studio. An associated time sequence diagram is also provided with each scenario to clarify the operation. It should be noted that the labels outside of the various blocks in FIGs. 5, 7, 9, 11, 13, and 15 denote physical studio components, whereas the labels within these blocks generally denote software objects.

Similarly, it should be noted that the labels in parenthesis () on top of the time sequence diagrams in FIGs. 6, 8, 10, 12, 14, 16 and 17 denote physical studio components, whereas the labels above the parenthesis () generally denote software objects. Furthermore, the shaded regions correspond to User to Network dialogues and ATM connection control as defined in ITU-T Recommendation Q.2391, B-ISDN Application Protocols for Access Signaling (indicated by the keyword Q.2931). Components with endpoints and those with short double arrows are involved in the dialogue. User to user (UU) dialogues are indicated by single-ended arrows. These dialogues represent a user protocol entity requesting the services of another user protocol entity. Preferably, a C++ style language is used. In the event that a path traverses multiple components, the UU dialogue line is shown only pointing to the final component.

It is recognized that not all components in the studio will be completely enabled to provide a distributed object oriented interface, e.g., a CORBA interface and to participate in communication dialogues with other studio CORBA-like objects. In those cases where a component does not provide this high level of control, the present command and control architecture uses a "proxy server" object. A proxy server object (or proxy object) is one which communicates with studio components on behalf of the component that it is representing. The object then communicates directly with the component using a low level communications protocol such as the Internet Engineering Task Force's (IETF) Simple Network Management Protocol (SNMP) or a proprietary communications protocol. The proxy object significantly enables an open architecture in the digital studio.

FIG. 5 illustrates a block diagram of an example of a DS-CC dialogue for controlling a proxy device (transmitter) 730. A proxy object 712 for the transmitter 730 is illustrated on the PTAS (server) 710. Although the proxy object 712 is illustrated as residing on the PTAS (server) 710, it is not so limited.

3NSDOCID: <WO__9811724A1_I_>

Instead, FIG. 5 illustrates the flexibility of the software object where it can reside on other studio components or platforms distinct from the physical device that it represents.

The Play-to-Air Control Workstation (PTACWS) (server) software client object 742 accesses the proxy object 712, which then communicates directly with the transmitter 730 through low level communication path 750, e.g., a RS 232 port. The actual ATM connections are constructed between the video source, the PTAS server 710 and the destination, transmitter 730, through the router 720 with a system resource manager (SRM), which is disclosed by the DSM-CC. The SRM is generally responsible for the network services, e.g., setting up bandwidth and communication channel between studio components or managing network resources (e.g., scheduling of network resources).

A DS-CC time sequence diagram for the scenario illustrated in FIG. 5 is provided in FIG. 6. FIG. 6 can be perceived as a method 800 for session management of a proxy component. FIG. 6 illustrates the interaction between the studio components (PTACWS, PTAS, Transmitter and Router/Network) in a time sequence diagram that progresses from top to bottom. It should be noted that the SRM is often referred to as the "network" and, as such, the network is perceived to reside in the router.

More specifically, in step 805, the client requests the network to setup a session. A session provides the logical relationship between two users or network applications, e.g., a studio component coming "on-line".

In step 810, the client attaches to the ServiceGateway, which is an indication that the client is announcing its intent to use studio components. The ServiceGateway is defined in the DSM-CC which can be perceived as an access point.

In step 815, the transmitter is located using the naming context as described in the directory structure of FIG. 4 above. The transmitter is found through the "Devices Directory/Transmitters Directory". Since FIG. 4 is a distributed naming context, when the "Open" specification is issued, method 800 may traverse over different points before locating the desired object. As such, since the Proxy Transmitter 712 resides on the PTAS, the arrow in step 815

BNSDOCID: <WO__9811724A1_I_>

ultimately points to the server of the PTAS.

Furthermore, step 815 comprises the operation of finding the appropriate device factory (using "Find"). As such, although the term "Open" is used in step 815, those skilled in the art will realize that it is in fact a "Find" specification.

In step 820, the client of method 800 opens the transmitter and receives an object (proxy object) representing the transmitter. It should be noted that when accessing the transmitter itself, the "Open" specification is a factory request, i.e., creating an instance of the device (using "Open"). When the studio device, transmitter, is opened, method 800 receives back a connection port, e.g., returning VCI/VPI to be used in future connections. Namely, when the client of method 800 opened a studio device, it effectively acquired control over this studio device. Thus, the dashed lines indicate a low level dialogue between the proxy object and the actual device.

In step 825, the Servers/PTAServer is located again using the naming context as described in the directory structure of FIG. 4 above, which, like step 815 above, comprises the operation of finding the appropriate server factory (using "Find"). However, although not specifically shown, step 825 also comprises the additional step of accessing the PTAS itself. Thus, the "Open" specification is again a factory request, i.e., creating an instance of the server (using "Open"). As such, although the term "Open" is used in step 825, those skilled in the art will realize that it is in fact a "Find" and "Open" specification. Furthermore, when the studio device, PTAS, is opened, method 800 also receives back a connection port, e.g., returning VCI/VPI to be used in future connections.

In step 830, method 800 uses an "ProxyOpen" specification to forward a video clip called "VideoClipOne" to a destination, which is the transmitter. Namely, method 800 is opening a path (video stream), illustrated as 760 in FIG. 5) from the PTAS 710 to the transmitter 730. A continuous feed session is presumed.

In step 835, a low level dialogue is established between the proxy transmitter and the server of the PTAS through the SRM, e.g., resource is requested by PTAS to proxy transmitter and proxy transmitter communicates with the transmitter. The dashed lines indicate a low level dialogue between the

NSDOCID: <WO___9811724A1_I_>

proxy object and the actual device. It should be noted that the "AddResource" connection request is an activity of the SRM.

In step 840, a direct connection is established, e.g., a Q.2931 or ATM connection from the PTAS to the transmitter managed by the SRM (720). In summary, a high level dialogue produced a direct connection being established between the PTAS and the transmitter. More importantly, method 800 now acquired a "stream" ("strm") which can be selectively controlled. It should be noted that "stream" is a software object that resides on the PTAS. Namely, "stream" is a software object that corresponds to the state of the underlying video clip. This software object can be accessed in a useful manner as discussed in steps 845 and 850.

In optional step 845, method 800 can selectively access and manipulate the object stream, e.g., play the stream at a prescribed time (6:00 PM, etc.).

In optional step 850, method 800 can selectively access and manipulate the object stream, e.g., stop the stream at a prescribed time. In fact, various VCR like functions are contemplated, e.g., rewind, pause or forward.

In step 855, the client detaches from the ServiceGateway, which is an indication that the client is announcing its intent to terminate access to studio components.

Finally, in step 860, the client may request the network to teardown a session. Thus, FIG. 6 illustrates a method 800 where a request to the proxy transmitter object results in the return of connection information so that connection establishment may proceed directly between the video source and the destination (transmitter). This general approach is used to advantage in more complex cases discussed below.

FIG. 7 illustrates a block diagram of an example of a DS-CC dialogue for controlling two proxy objects in a digital studio embodiment in which neither the source nor the destination is directly controlled. Namely, the software client object 922 (e.g., a playlist-1 object) on the PTACWS 920 (server) simply accesses the proxy object livefeed 912 and proxy object transmitter 914 to effect the transfer of a video stream from a source (live-feed) 930 to a destination (transmitter) 950 via a router 940. Similar to the example of FIG. 5, the proxy

BNSDOCID: <WO__9811724A1_I_>

objects (proxies) then communicate directly with the livefeed component 930 and transmitter 950 through low level communications paths 970. In this example, both the source and destination have proxies and both proxies reside on the studio component PTAS.

A DS-CC time sequence diagram for the scenario illustrated in FIG. 7 is provided in FIG. 8. FIG. 8 can be perceived as a method 1000 for session management of two proxy components. FIG. 8 illustrates the interaction between the studio components (PTACWS, PTAS, LiveFeed, Transmitter and Router/Network) in a time sequence diagram that progresses from top to bottom.

More specifically, in step 1005, the client requests the network to setup a session. In step 1010, the client attaches to the ServiceGateway, which is an indication that the client is announcing its intent to use studio components.

In step 1015, method 1000 opens the transmitter 950 and gains control of the transmitter. Although step 1015 illustrates the opening of the transmitter in a single step, it is simply a shortcut expression to indicate the similar steps of 815 and 820 as discussed above in FIG. 6.

In step 1020, method 1000 opens the LiveFeed 930 and gains control of the LiveFeed. In fact, although not shown, step 1015 also includes the step of opening LiveFeed which is also similar to 815 and 820 as discussed. The initial Open only reserves the Transmitter resources while the second Open actually causes a connection to be created between the source and destination. Again, the second "Open" specification is used broadly here. A connection is generally created when operation CreateStream (similar to the steps 825 and 830 of FIG. 6) is performed. Namely, those skilled in the art will realize that the CreateStream operation comprises the steps associated with opening a path between a source and a destination as discussed above. The proxy feed will initiate a connection request with the SRM using a continuous feed session.

In steps 1022 and 1024, the connection is opened between the livefeed and the transmitter by the SRM, for example using a pair of ITU-T Q.2931 Call Setup Indications issued from the switch. This results in a DSM-CC session Setup Indication at the Proxy Transmitter object and a response to the Proxy Live Feed object. The status of this activity is then returned to the initiator by the Proxy

3NSDOCID: <WO__9811724A1_I_>

LiveFeed object.

Similarly, optional steps 1025 and 1030 also illustrate the availability of VCR like functions in this example. Finally, in step 1035, the client detaches from the ServiceGateway, and the client may request the network to teardown the session in step 1040. The underlying theme is again reiterated where the source and destination paths are specified, thereby resulting in the establishment of a communications dialogue.

In summary, the original command originated from the client object and went to the proxy LiveFeed, which effected a connection with the LiveFeed component. The Proxy LiveFeed then initiates a dialogue with SRM to establish a connection between the physical transmitter and the physical LiveFeed, thereby resulting in a low level ATM connection 960 between the transmitter and the LiveFeed.

FIG. 9 illustrates a block diagram of an example of a DS-CC dialogue for controlling two proxies with video stream passing through a transcoder in a digital studio embodiment in which neither the source nor the destination is directly controlled. Namely, the software client object 1122 on the PTACWS 1120 (server) simply accesses the proxy object livefeed 1112 and proxy object transmitter 1114 to effect the transfer of a video stream from a source (live-feed) 1130 to a destination (transmitter) 1150 via a transcoder 1180 and router 1140. The proxies communicate directly with the livefeed component 1130 and transmitter 1150 through low level communications paths 1170. In this example, both the source and destination have proxies and both proxies reside on the studio component PTAS.

Furthermore, the video path 1135 from the livefeed now passes through a transcoder 1180 which is assumed to perform some task such as frame rate conversion, aspect ratio conversion , interlace to progressive scan conversion or bit rate conversion. The video path 1160 from the transcoder then passes through the router and arrive at the transmitter. The transcoder 1180 is controlled by a proxy object 1190 which resides on a studio component, which is the router 1140 in this example.

The DS-CC time sequence diagram for the services depicted in FIG. 9 is

BNSDOCID: <WO___9811724A1_I_>

shown in FIG. 10. To reduce repetitive description, description for steps 1205, 1210, 1225, 1230, 1240, 1245, 1250, 1260 and 1265 is referred to the above discussion with regard to FIGs. 6 and 8.

More specifically, as shown, each controlling object is responsible for obtaining those component resources (filters, switchers, etc.) as are needed prior to actually establishing communication connections. In this example, the transcoder is first obtained in step 1215 (by the Open command) and then a connection is made from the Transmitter to the Transcoder in step 1220 and from the video server to the transcoder in step 1235. Each of these paths is considered to be a separate connection (i.e. there are two "calls" to the transcoder, one "to" it and one "from" it. Once a path has been established, the video is controlled as described earlier with the additional function performed by the transcoder in step 1255. Notice that all connections are managed by the SRM, therefore the SRM has complete knowledge of the system topology. A resource oracle object may be attached to the SRM to provide topology information for studio management by authorized users.

FIG. 11 illustrates a block diagram of an example of a DS-CC dialogue for controlling two proxies with video stream passing through a Switcher 1380 in a digital studio embodiment. Namely, the software client object 1322 on the PTACWS 1320 (server) simply accesses the proxy transmitter object 1312, PPS server 1330, PTAS server 1310 and the PTASW (switcher) 1380 to effect (or splicing) two streams together to form a new stream. In this context it is assumed that a switcher 1380 (splicer) exists that takes two or more input streams 1370 and 1375 and is able to switch between them under external control through a router 1340 to the transmitter via path 1360. In the embodiment shown in FIG. 11, a video stream is composed from two sources, PTAS 1310 and PPS 1330. The destination is a Transmitter 1350 which is controlled by a proxy object 1313 which resides on the PTAS 1310. The client (controlling object 1322) resides on PTACWS 1320.

The DS-CC time sequence diagram for the services depicted in FIG. 11 is shown in FIG. 12. To reduce repetitive description, description for steps 1405, 1410, 1425, 1430, 1440, 1445, 1455, 1460, 1465, and 1470 is referred to the above

INSDOCID: <WO___9811724A1_I_>

discussion with regard to FIGs. 6, 8 and 10.

FIG. 12 provides a time sequence diagram for a typical scenario for the controlling object (e.g., a PlayList-1 object). As discussed above, each component resource is first obtained using an "Open" command, and then the connections are made between itself and the other objects that will be involved in the video sequence. In this example, the switch 1380 is the first studio component to be opened in step 1415. Method 1400 then opens transmitter 1420, source 1 and source 2 in steps 1420, 1435 and 1450 respectively.

The control sequence at the end of the time sequence diagram illustrates how it is the responsibility of the controlling object to compute any delays that might be involved in the video path. First stream strm-1 in step 1465 is commanded to play (presumably before 6:00 pm), the output to the transmitter 1350 is therefore strm-1. At 6:00 pm - 10msec a command is sent to stream strm-2 in step 1470 to start playing at 6:00 pm - 5 msec. It is assumed that the delay incurred in this command being delivered is less than 5 msec. The time for the first bit of the video stream to reach the switch 1380 is assumed to be 5 msec, therefore the start of the first frame arrives at the switcher at 6:00 pm exactly. The command to switch at 6:00 pm in step 1475 is sent to the switcher some period before the switch is required, in this case at 6:00 pm - 5 msec. The switch occurs within the switcher 1380 at 6:00 pm and the resulting stream is derived from stream strm-2. At this point strm-1 may be terminated.

FIG. 13 illustrates a block diagram of an example of a DS-CC dialogue for storing video from a server 1510 to a client 1520 in a digital studio embodiment. The storage is done under client control via a router 1530. One important requirement within the context of a compressed digital studio is to store streams, for example, a new stream constructed from a series of edits may require storing back on a video server as a new object. FIG. 13 illustrates the storing of video from a server (the PPS) 1510 to a new storage object on the PTAS 1520.

The DS-CC time sequence diagram for the services depicted in FIG. 13 is shown in FIG. 14. To reduce repetitive description, description for steps 1605, 1610, 1630, 1635, 1640, 1645, 1650, and 1655 is referred to the above discussion with regard to FIGs. 6, 8, 10 and 12.

BNSDOCID: <WO__9811724A1_I_>

Notice that the new storage object must first be created (using "New") in step 1615 by method 1600. The object is then opened in step 1620 using a new construct called "OpenWrite" which returns connection information for this object. A stream is then created in the usual manner in step 1625. In this example copying takes place at normal play time, however, the Play and PlayAt commands may be extended to permit faster (or slower) than real-time transmission.

FIG. 15 illustrates a block diagram of an example of a DS-CC dialogue for the distribution of digital video streams to multiple recipients and shared clients via a router 1740. In the embodiment shown in FIG. 15, Client1 1722 on PTACWS 1720 initially opens a video feed to two destinations, a Transmitter 1750 via proxy transmitter 1712 and an archive backup 1730 PPS. Since these two video feeds are derived from the same source they cannot be separately controlled. At some later point, client2 1714 (subject to authorization) also attaches to the video stream 1742 and directs the output to a monitor 1760 via proxy monitor 1716. In the embodiment shown, client2 cannot control the stream 1742, although client1 can.

The DS-CC time sequence diagram for the services depicted in FIG. 15 is shown in FIG. 16. To reduce repetitive description, description for steps 1805, 1810, 1815, 1820, 1825, 1830, 1845, 1850, 1855, 1864, 1866 and 1868 is referred to the above discussion with regard to FIGs. 6, 8, 10, 12 and 14.

As can be seen, on the occasion that client1 opens the second destination in step 1835, a specification is given to indicate that the connection is to augment an existing one via some parameter mechanism. In FIG. 15, this is indicated by the parameter Mode=LeafJoin in step 1840 by method 1800. This would be manifested as a Q2931 Call SetupIndication to the video server with the appropriate connection information. It should also be pointed out that there are a number of ways of opening the second stream. One way is by specifying the video source (indicated in the figure by the first occurrence of C1:LeafJoin). Another approach is to specify one of the destination components (indicated in the figure in the shaded region in step 1860 at the bottom). In this case the SRM will identify which source is supplying the video stream and initiate a join from this stream. In either case the resulting stream emanates from the source and is split within

BNSDOCID: <WO__9811724A1_I_>

the switch. This is important for performance considerations.

Another expected studio activity is for a second client to require access to a previously opened stream. This case is illustrated for the case of client2 in FIG. 17. To reduce repetitive description, description for steps 1905, 1910, 1925, and 1930 is referred to the above discussion with regard to FIGs. 6, 8, 10, 12, 14 and 16.

The client starts with a usual SessionSetup and an Open to the destination proxy in step 1915 (a Monitor in this case). The proxy is then linked into the stream by using the Mode=LeafJoin indication in step 1920. In this case the source name for the stream has been specified.

A command and control architecture for a digital studio having a plurality of studio components is described. The architecture has means for locating one of the studio component, means, coupled to said locating means, for acquiring control of said studio component; and means, coupled to said acquiring means, for specifying an interconnection to said studio component under control to establish a stream through said studio component. The locating means can locate an object, such as a proxy object, corresponding to said studio component.

The locating can be done using a naming context with an hierarchical directory structure, where said hierarchical directory structure includes a service gateway at a root level. The stream can be an object corresponding to a state of the underlying digital data and can be selectively manipulated to effect real-time control of studio activities.

The established stream can support multiple clients, multiple recipients and storage of digital data and it can be spliced from other streams within the digital studio.

Also described is a method for managing digital data for a digital studio having a plurality of studio components, using the above-specified means.

An original control methodology for typical studio operations such as splicing, storing and transcoding is disclosed.

It is to be understood that the apparatus and method of operation taught herein are illustrative of the invention. Modifications may readily be devised by those skilled in the art without departing from the spirit or scope of the invention.

BNSDOCID: <WO__9811724A1_I_>

What is claimed is:

1. A command and control architecture for a digital studio having a plurality of studio components, comprising:

means for locating one of the studio component;

means, coupled to said locating means, for acquiring control of said studio component; and

means, coupled to said acquiring means, for specifying an interconnection to said studio component under control to establish a stream through said studio component.

2. The command and control architecture of claim 1, wherein said locating means locates an object corresponding to said studio component.

3. The command and control architecture of claim 2, wherein said locating means locates said object using a naming context with an hierarchical directory structure, where said hierarchical directory structure comprises a service gateway at a root level.

4. The command and control architecture of claim 1, wherein said stream is an object corresponding to a state of the underlying digital data.

5. The command and control architecture of claim 1, wherein said established stream is spliced from other streams within the digital studio.

6. A method for managing digital data for a digital studio having a plurality of studio components, comprising the steps of:

locating one of the studio component;

acquiring control of said studio component; and

specifying an interconnection to said studio component under control to establish a stream through said studio component.

3NSDOCID: <WO___9811724A1_I_>

7. The method of claim 6, wherein said locating step locates an object corresponding to said studio component.

8. The method of claim 7, wherein said object is a proxy object.

9. The method of claim 6, wherein said stream is an object corresponding to a state of the underlying digital data.

10. The method of claim 9, wherein said stream is selectively manipulated to effect real-time control of studio activities.

11. A studio class directory structure in a computer-readable media, comprising:

a gateway service interface as a root directory;

a devices class directory, coupled to said root directory;

a servers class directory, coupled to said root directory;

a studios class directory, coupled to said root directory; and

a software resources class directory, coupled to said root directory.

12. The studio class directory structure of claim 11, further comprising:

a livefeeds directory, coupled to said devices class directory;

a filters directory, coupled to said devices class directory; and

a transmitters directory, coupled to said devices class directory.

13. The studio class directory structure of claim 12, further comprising:

a switchers directory, coupled to said filters class directory; and

a transcoders directory, coupled to said filters class directory.

14. The studio class directory structure of claim 11, further comprising:

a Play-To-Air server directory, coupled to said servers class directory;

a Production Program server directory, coupled to said servers class directory; and

:NSDOCID: <WO___9811724A1_I_>

a Network Program server directory, coupled to said servers class directory.

5

20

3NSDOCID: <WO___9811724A1_I_>

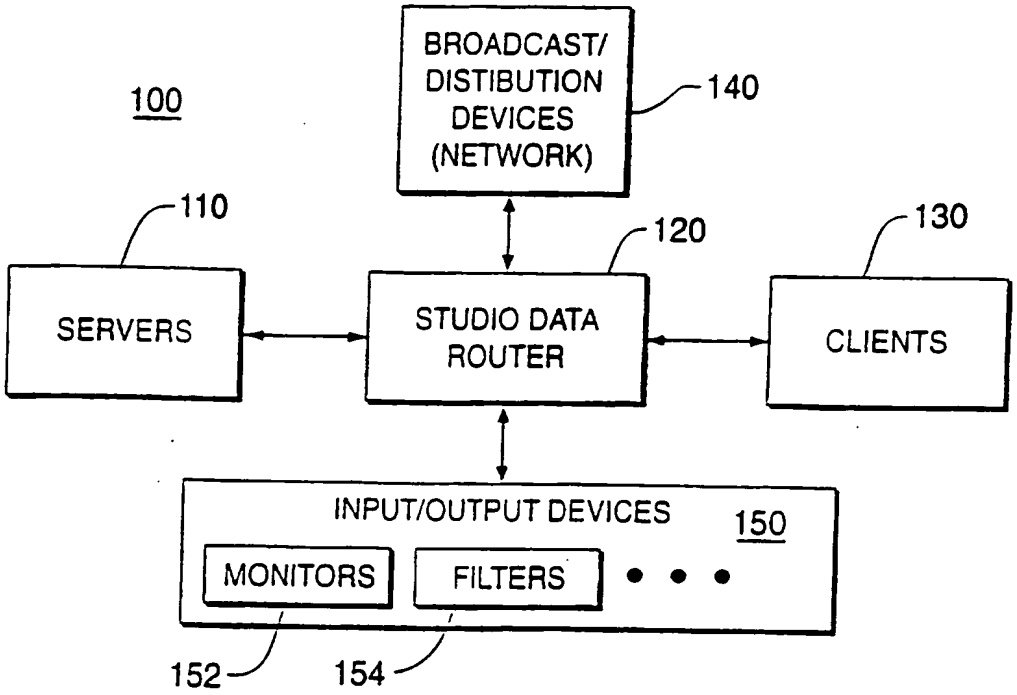


FIG. 1

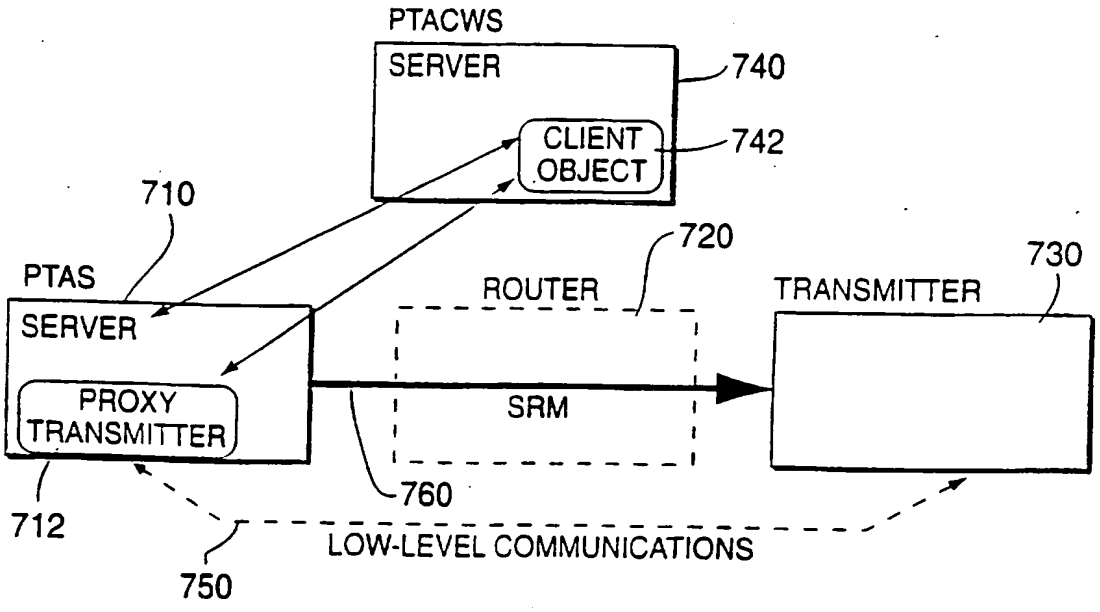
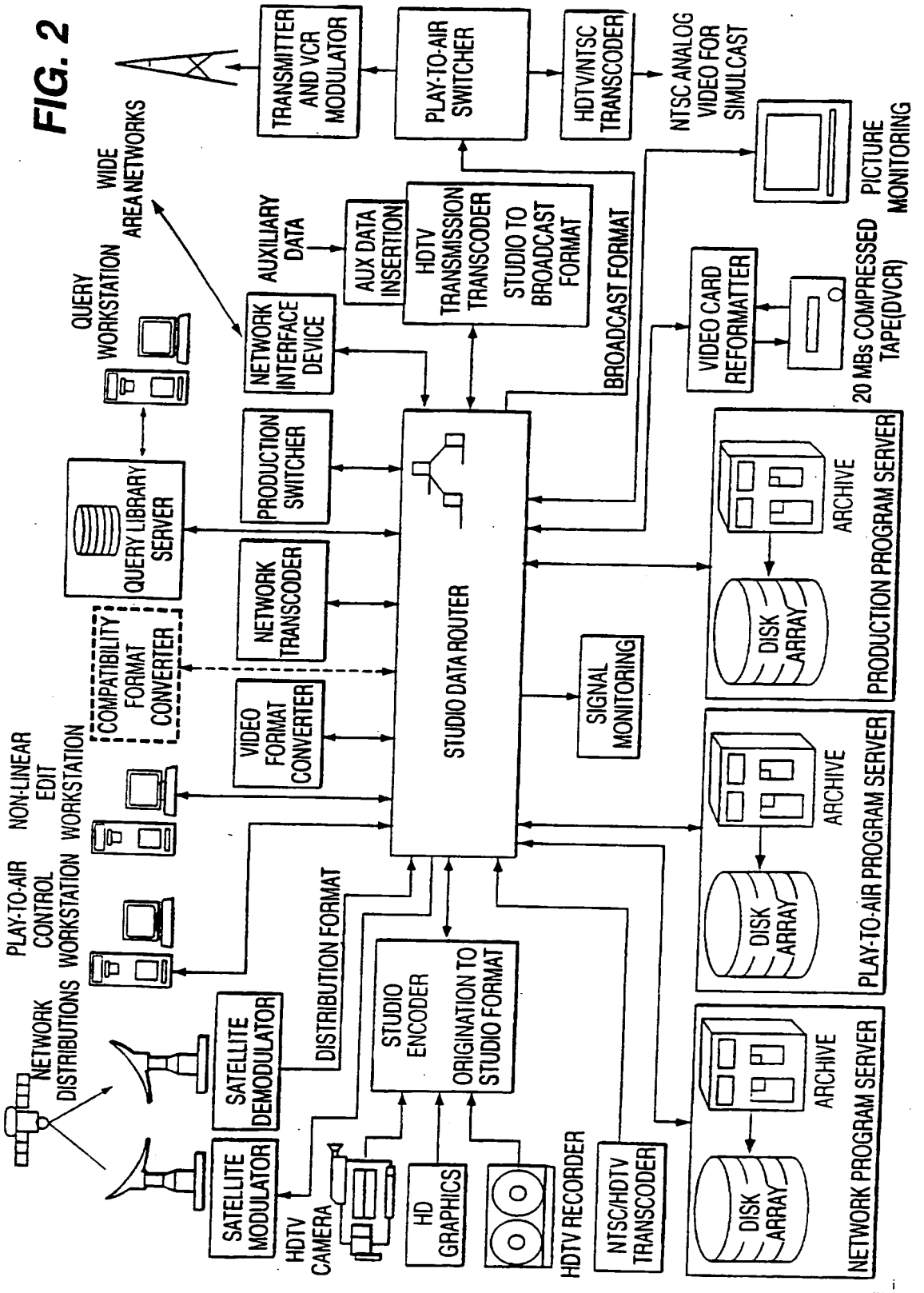


FIG. 5

3NSDOCID: <WO___9811724A1_I_>

FIG. 2



INSDOCID: <WO___9811724A1_I_>

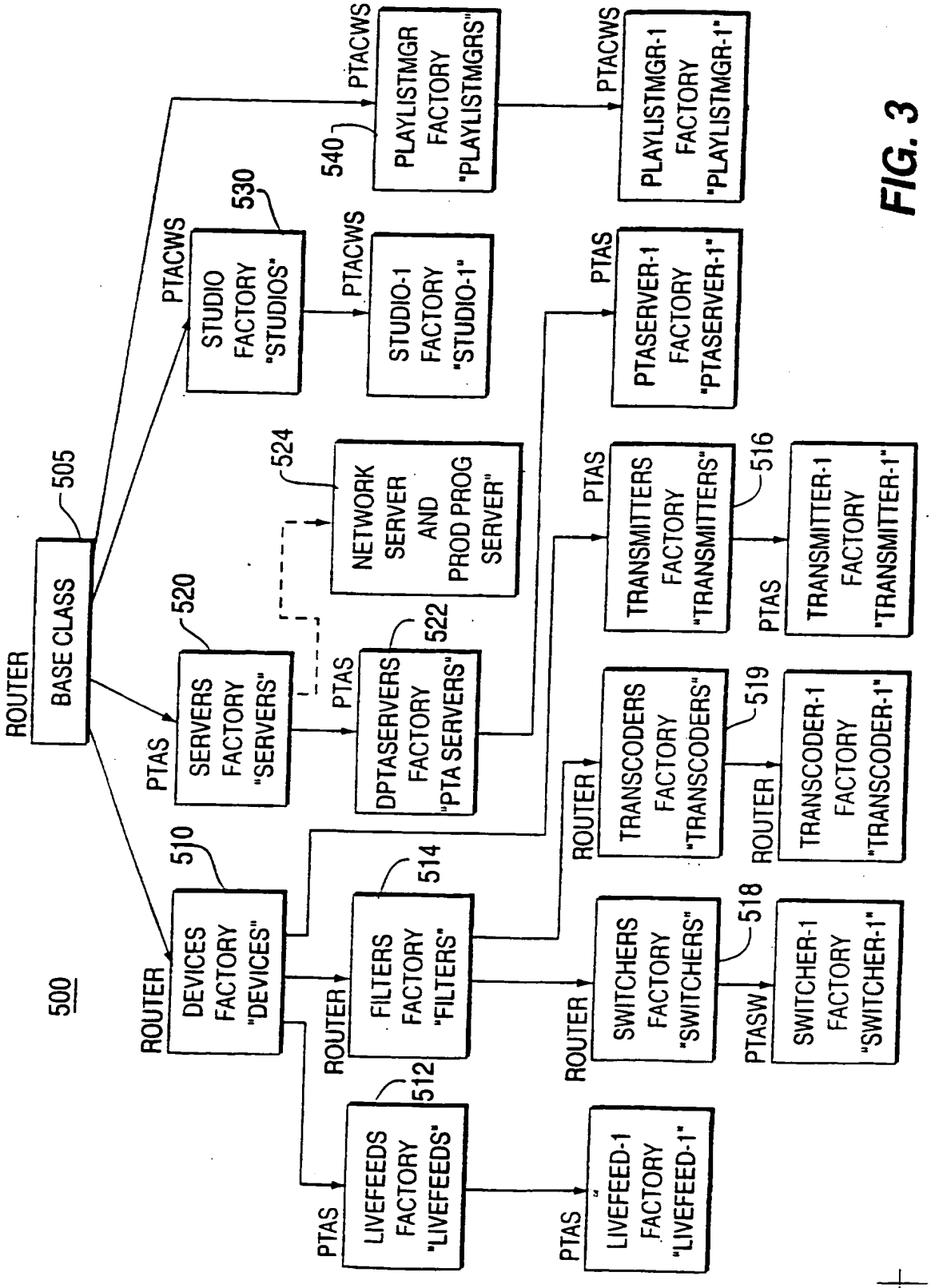


FIG. 3

INSDOCID: <WO___9811724A1_I_>

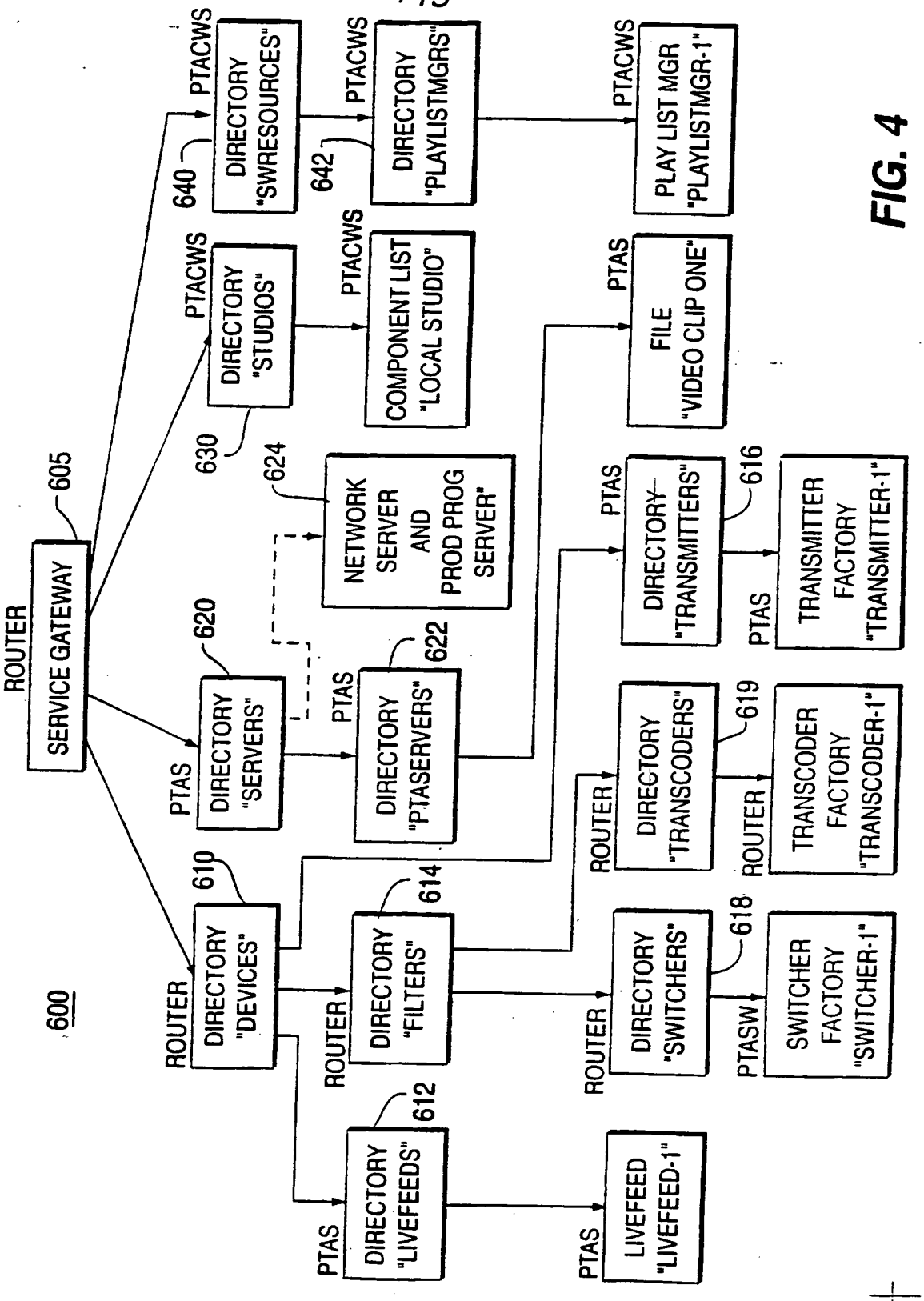


FIG. 4

NSDOCID: <WO___9811724A1_I_>

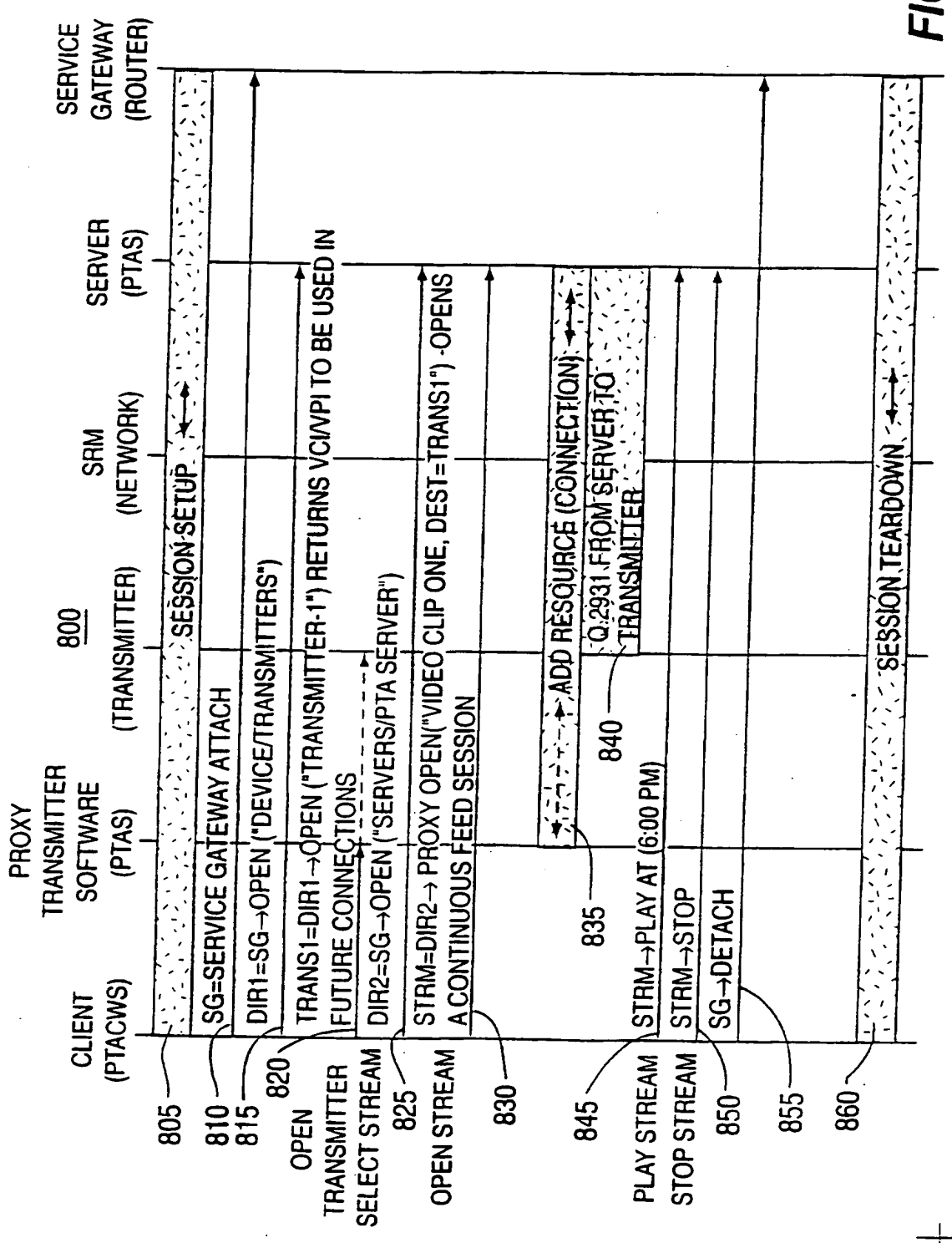


FIG. 6

BNSDOCID: <WO___9811724A1_I_>

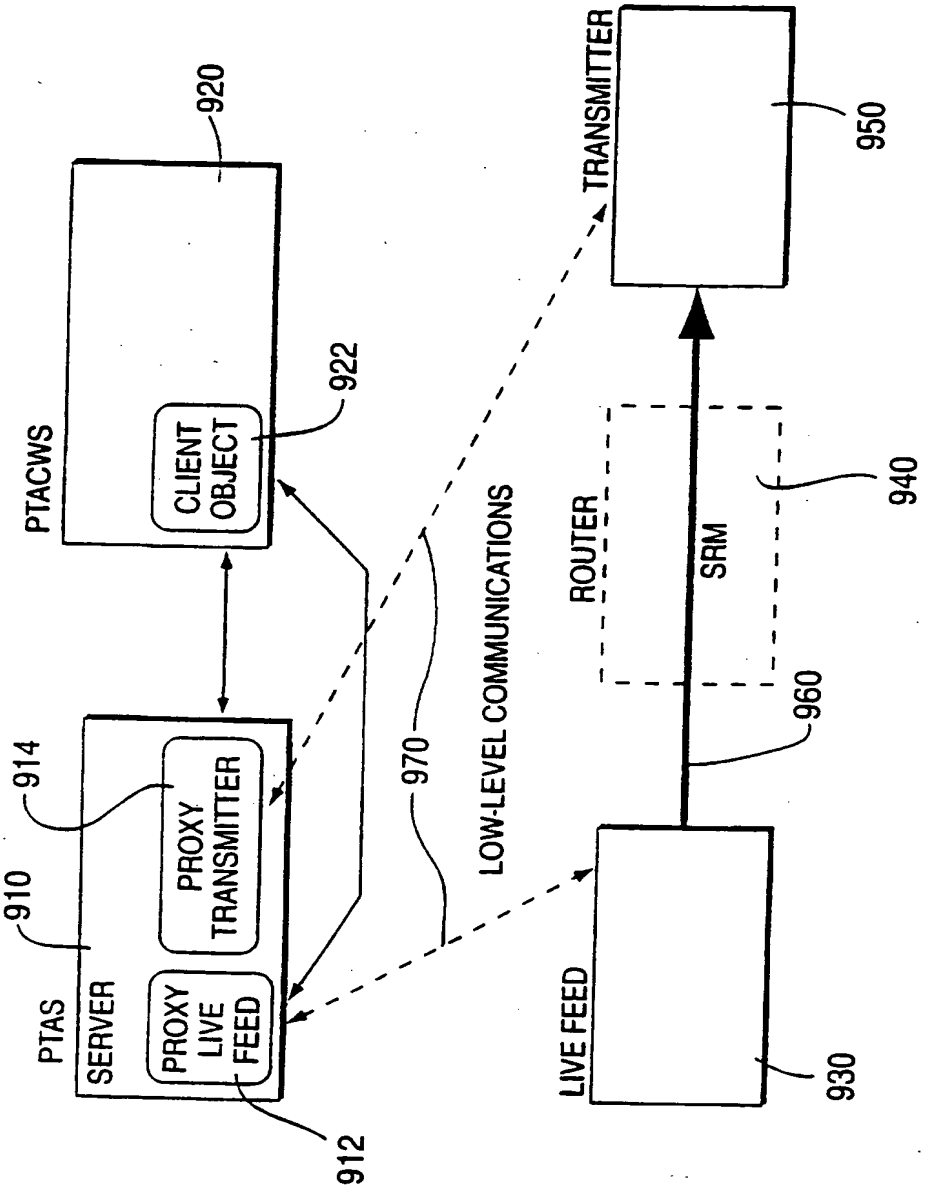


FIG. 7

NSDOCID: <WO___9811724A1_I_>

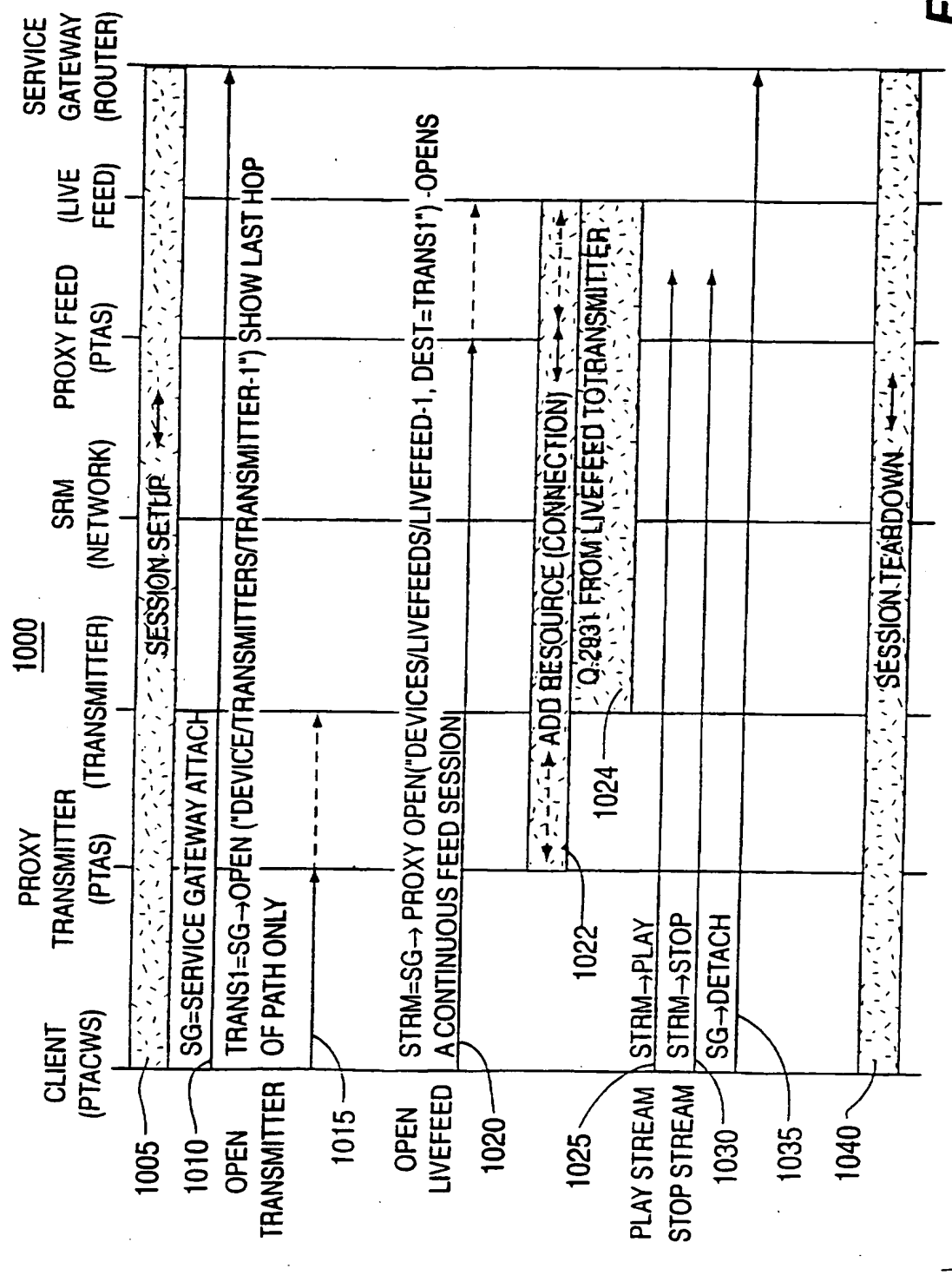


FIG. 8

BNSDOCID: <WO___9811724A1_I_>

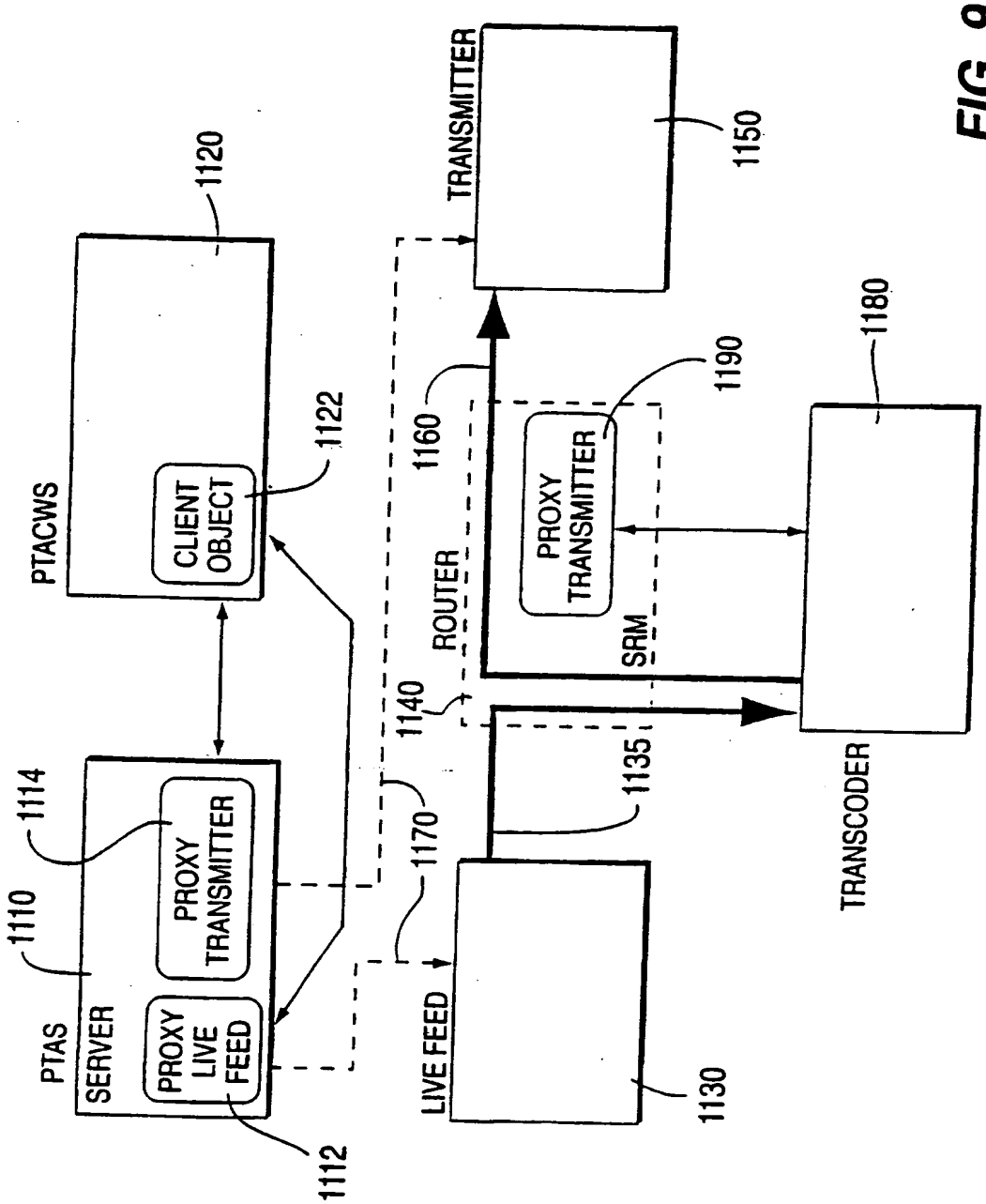


FIG. 9

NSDOCID: <WO___9811724A1_I_>

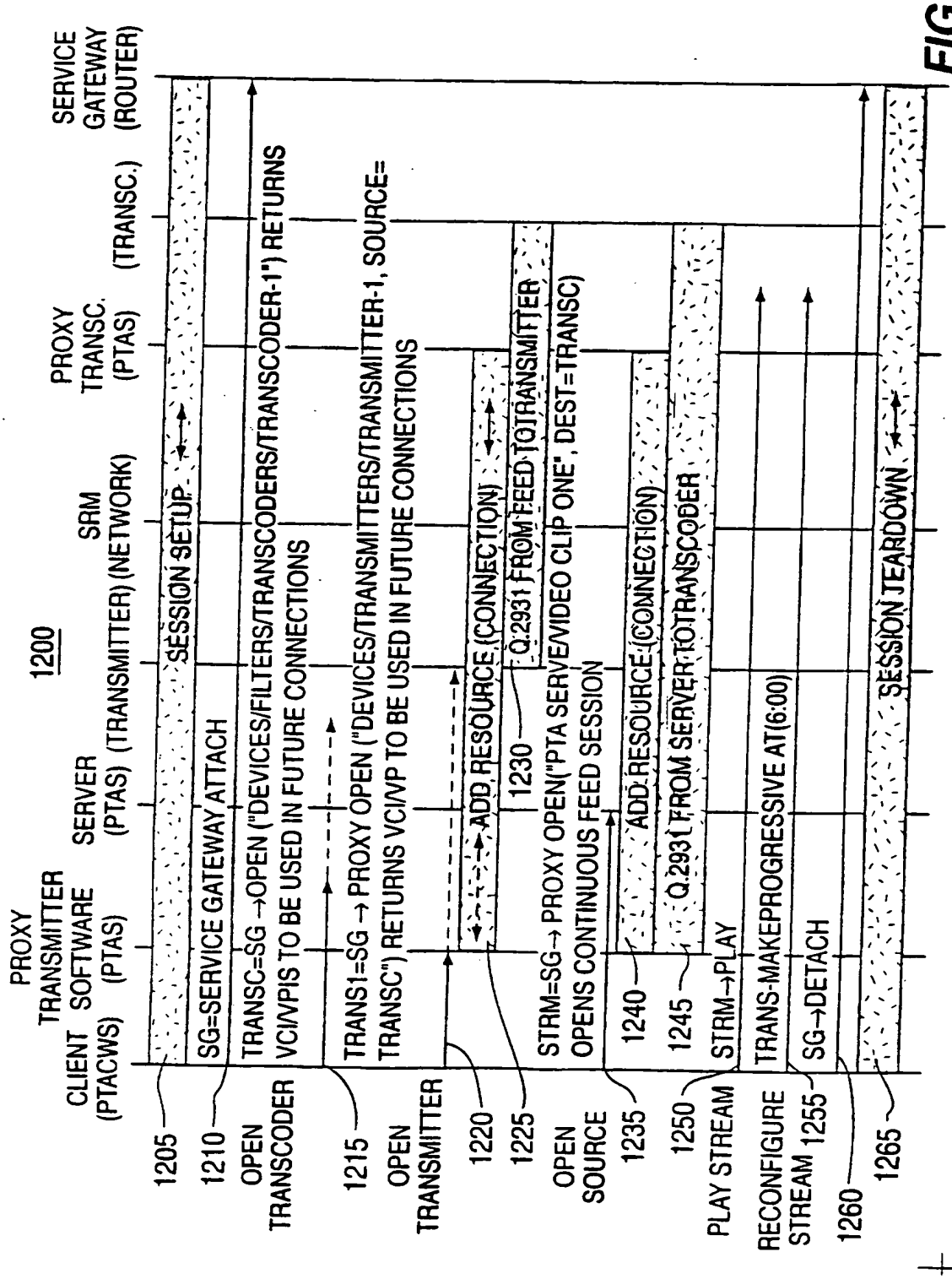


FIG. 10

BNSDOCID: <WO___9811724A1_I_>

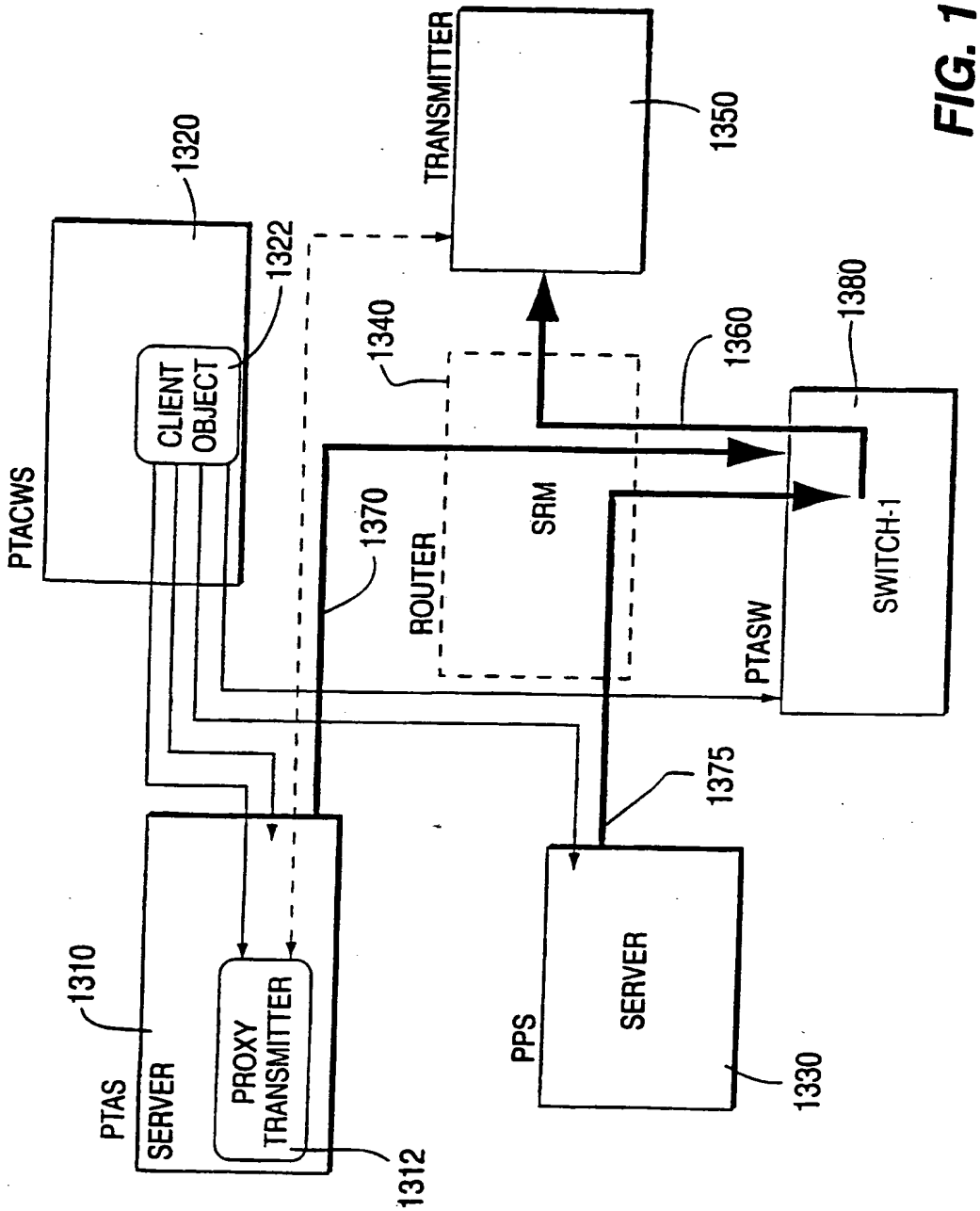


FIG. 11

NSDOCID: <WO___9811724A1_I_>

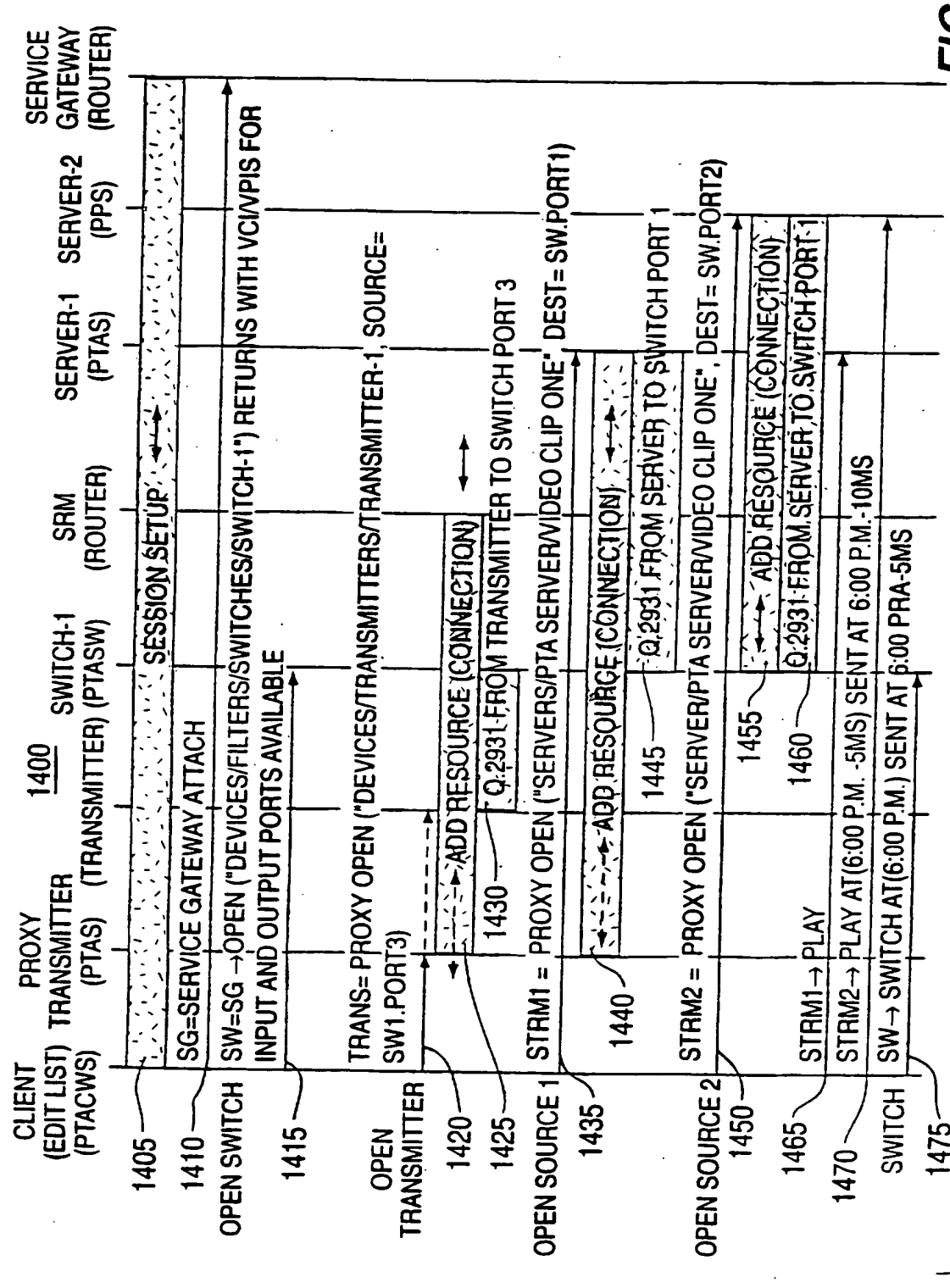


FIG. 12

BNSDOCID: <WO__9811724A1_I_>

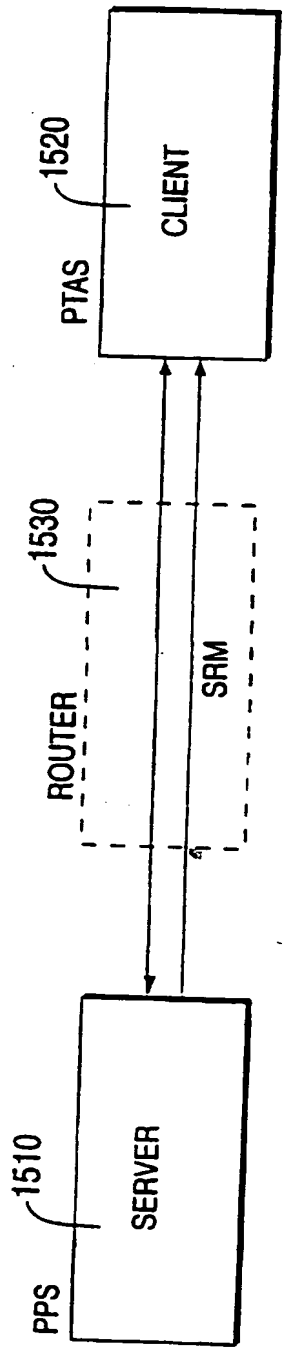


FIG. 13

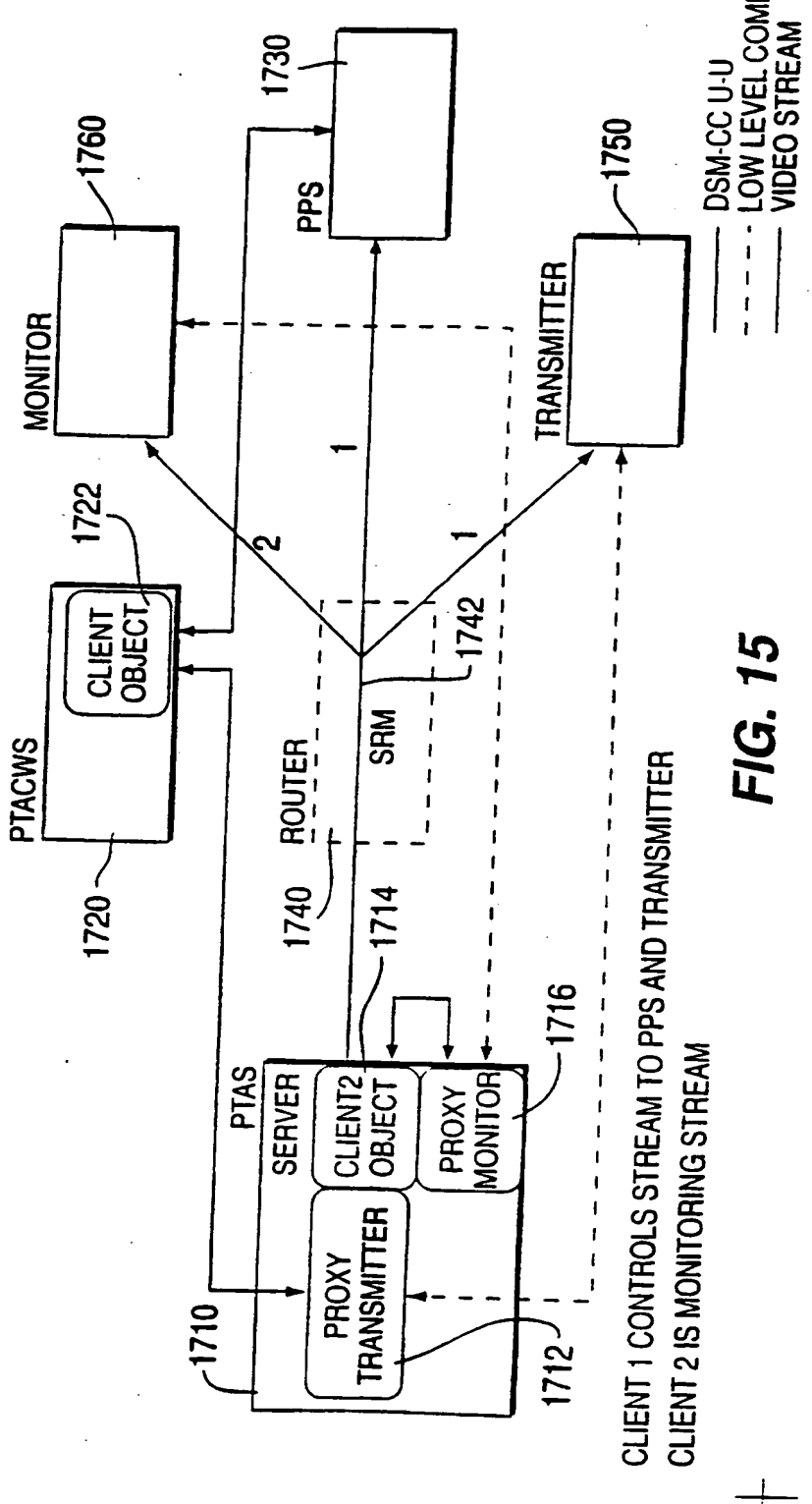
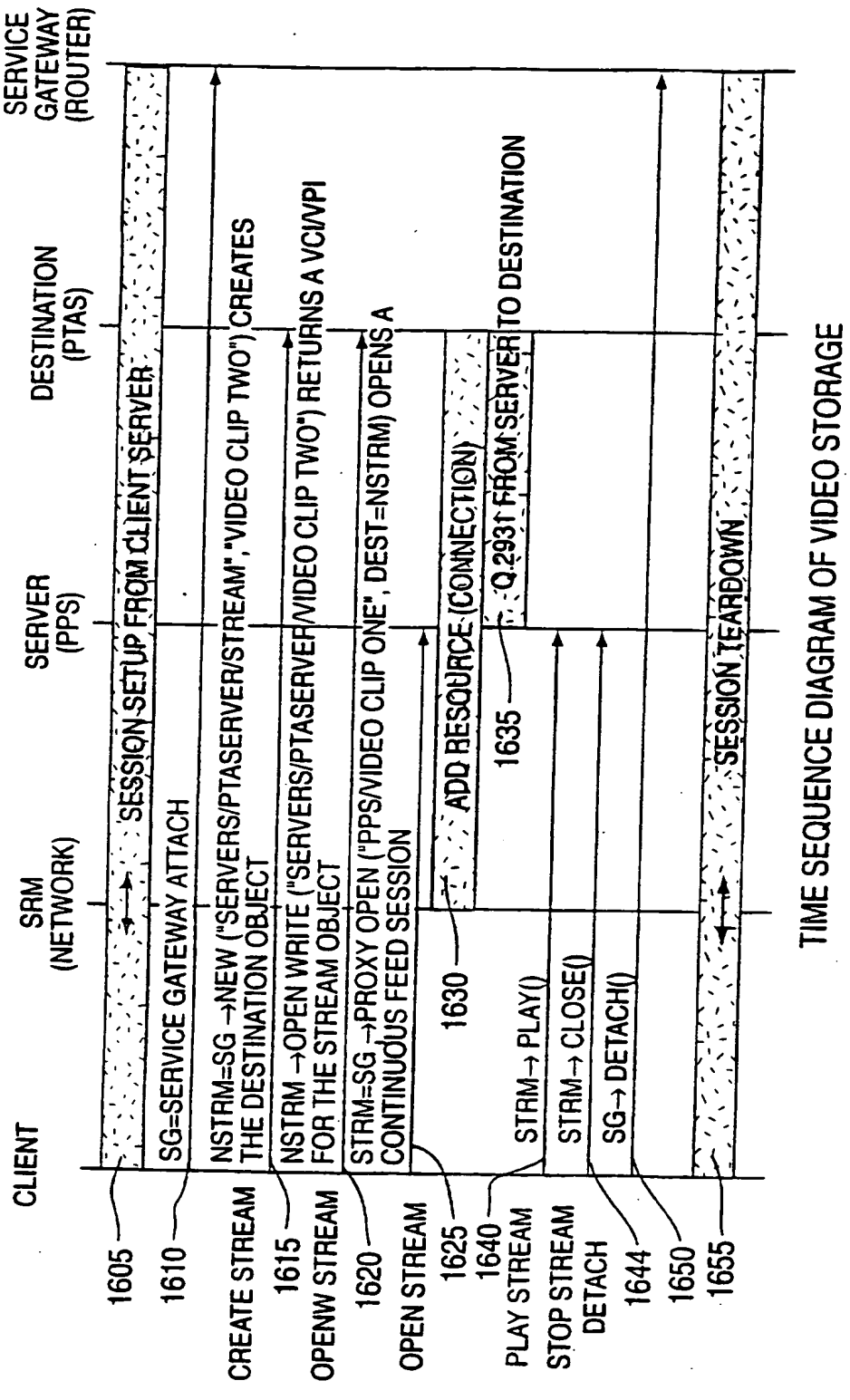


FIG. 15

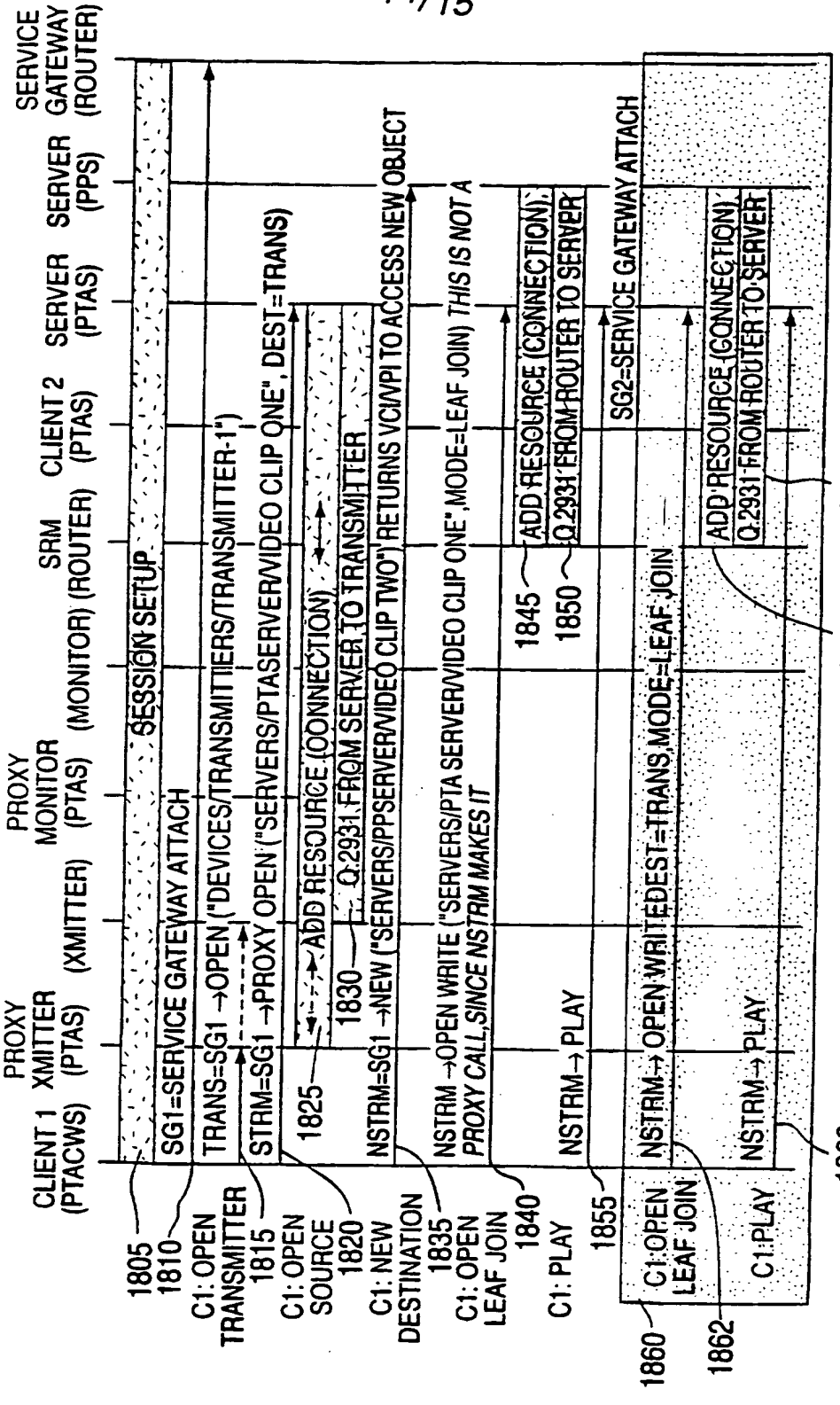
3NSDOCID: <WO___9811724A1_I_>



TIME SEQUENCE DIAGRAM OF VIDEO STORAGE

FIG. 14

BNSDOCID: <WO___9811724A1_I_>



TIME SEQUENCE DIAGRAM FOR MULTIPLE RECIPIENTS.

FIG. 16

3NSDOCID: <WO___9811724A1_I_>

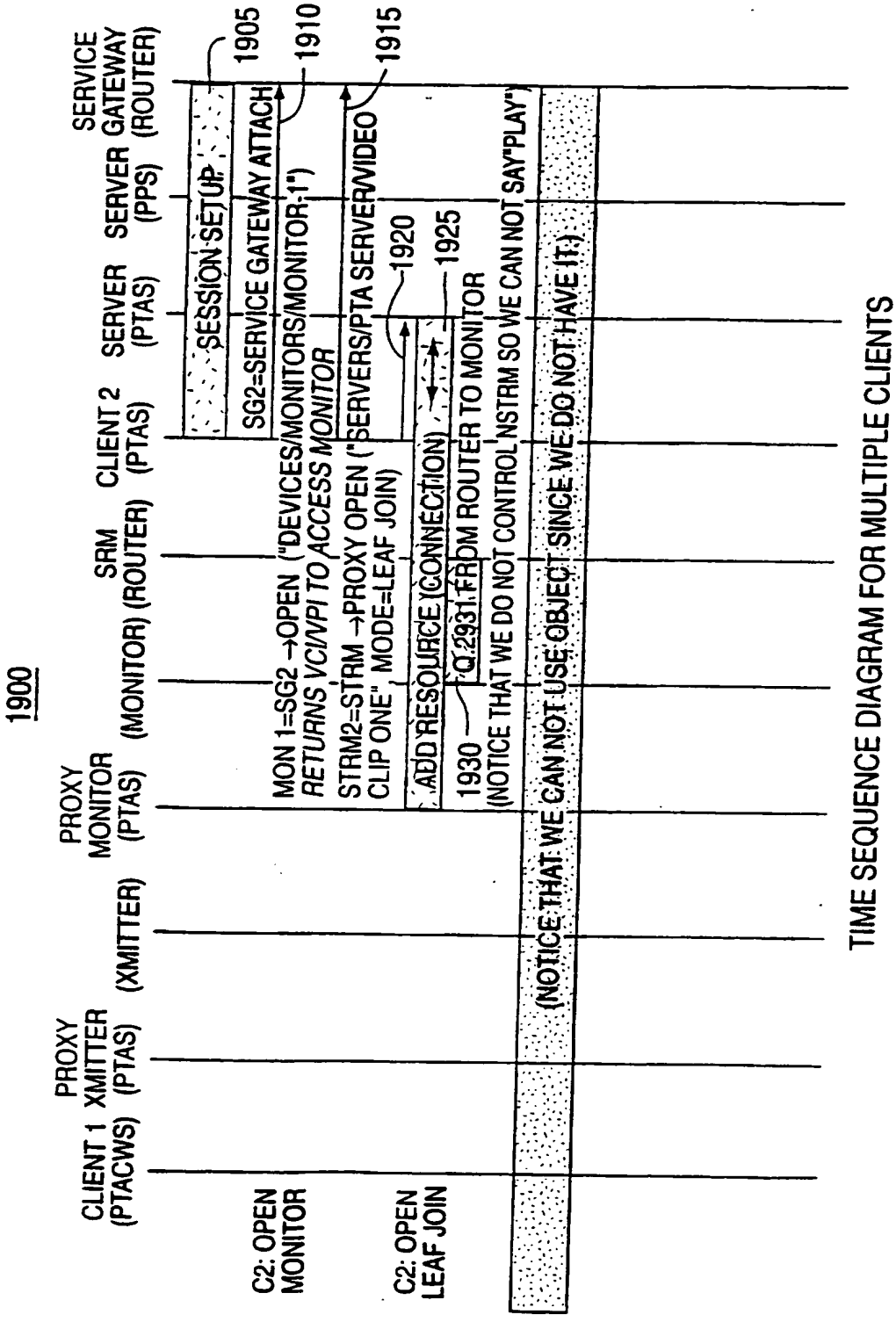


FIG. 17

BNSDOCID: <WO___9811724A1_I_>

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/17111

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :H04N 7/14
US CL :348/6, 12; 455/3.1, 5.1, 6.3; 345/335, 356; 707/205
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 348/6, 7, 12, 13, 552; 455/3.1, 4.1, 4.2, 5.1, 6.1, 6.3; 345/146, 335, 352, 353, 356; 707/200, 205

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

NONE

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

NONE

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X,P	US 5,649,131 A (ACKERMAN ET AL) 15 July 1997, fig. 6, col. 9, line 19 to col. 10, line 16.	11, 13, 14
X	US 5,297,249 A (BERNSTEIN ET AL) 22 March 1994, figures 11-15 and col. 13, lines 10-23.	11-14
A	US 5,479,656 A (RAWLINGS, III) 26 December 1995, abstract and fig. 10.	11-14
X,E	US 5,671,225 A (HOOPER ET AL) 23 September 1997, figs. 2 & 3 and col. 3, lines 33-55 & 66-67 and col. 4, lines 1-7 & 42-59.	1-10
A	US 5,208,665 A (MCCALLEY ET AL) 04 May 1993 figure 5, abstract and col. 9, lines 12-34.	2-4, 7-9

[X] Further documents are listed in the continuation of Box C. [ ] See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"B"	earlier document published on or after the international filing date	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"&"	document member of the same patent family
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 04 DECEMBER 1997	Date of mailing of the international search report 03 FEB 1998
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer JEFFREY MURRELL Telephone No. (703) 305-8155

Form PCT/ISA/210 (second sheet)(July 1992)*

BNSDOCID: <WO___9811724A1_I_>

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/17111

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5,247,347 A (LITTERAL ET AL) 21 September 1993, abstract and figures 1-2.	1-2, 4-10

Form PCT/ISA/210 (continuation of second sheet)(July 1992)*

BNSDOCID: <WO___9811724A1_I_>

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/17111

BOX II. OBSERVATIONS WHERE UNITY OF INVENTION WAS LACKING

This ISA found multiple inventions as follows:

Group I, claim(s) 1-10, drawn to the locating and acquiring of digital data from a component in a digital studio, classified in class 348, subclass 7.

Group II, claim(s) 11-14, drawn to the structure of computer-readable class directories, classified in class 707, subclass 200.

The inventions listed as Groups I and II do not relate to a single inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons: the special technical features of Groups I. and II. are shown not to be corresponding, e.g. in the instant case, Group I has means for locating, means for acquiring, and means for specifying and Group II has a gateway service interface and different types of class directories. Therefore, the features do not correspond to form a single inventive concept.

INSDOCID: <WO___9811724A1_I_>